AMENDMENTS TO THE CLAIMS

1. (Currently amended) A method for processing program instructions comprising the steps of:

receiving at least one non-native instruction containing data representing operational code and ~~data representing~~ at least one non-native instruction flag ~~modification enable bit~~;

emulating the at least one non-native instruction[s] using at least one native instructions ~~containing the~~ including a flag modification enable bit;

determining whether the at least one flag modification enable bit allows updating of the at least one non-native instruction flag in response to emulating the at least one non-native instruction ~~executing the operational code~~; and

updating the at least one non-native instruction flag in response to determining ~~a status of~~ whether the at least one flag modification enable bit allows updating of the at least one non-native instruction flag.

2. (Original) The method of Claim 1 wherein the at least one instruction is at least one of:

an integer instruction, an arithmetic instruction and a logical instruction.

3. (Original) The method of Claim 1 wherein the at least one instruction includes at least one input operand.

4. (Currently Amended) The method of Claim 1 wherein the step of updating the at least one non-native instruction flag in response to determining ~~a status of~~ whether the at least one flag modification enable bit allows updating of the at least one non-native instruction flag includes updating a flag register if the flag modification enable bit is set to allow modification of a flag in the flag register.

5. (Currently Amended) The method of Claim 1 including the steps of:

providing a variable length instruction emulator that uses fixed length native instructions ~~as the at least one instruction,~~ to emulate variable length instructions; and

evaluating the flag modification enable bit to preserve flag bit settings for variable length instructions that are emulated using the fixed length native instructions.

6. (Currently Amended) The method of Claim 1 whercin the at least one native instruction is a fixed length instruction.

7. (Cancelled) The method of Claim 1 including the step of emulating non-native instructions using native instructions containing the flag modification bit.

8. (Original) The method of Claim 1 wherein the non-native instructions are variable length X86 instructions.

9. (Original) The method of Claim 8 including the steps of:

converting variable length X86 instructions to a plurality of native instructions wherein the plurality of native instructions include the at least one flag modification enable bit set to allow changing of non-native instruction flags in response to execution of the plurality of native instructions; and

emulating unconverted variable length X86 instructions using a plurality of native instructions wherein the native instructions include the at least one flag modification enable bit set to prevent changing of non-native instruction flags in response to execution of the plurality of native instructions.

10. (Currently amended) An apparatus for processing program instructions comprising:

a buffer coupled to receive at least one non-native instruction containing data representing operational code and ~~data representing~~ at least one non-native instruction flag ~~modification enable bit~~;

a variable length instruction emulator that ~~uses fixed length native instructions as the at least one instruction, to emulate variable length instructions wherein the variable length instruction emulator~~ emulates non-native instructions using at least one native instructions containing the flag modification enable bit; and

a controller operatively responsive to the at least one instruction stored in the buffer, that determines whether the at least one flag modification enable bit allows updating of the at least one non-native instruction flag in response to executing the operational code and ~~that updates at least one flag~~ in response to determining whether ~~a status of~~ the at least one flag modification enable bit allows updating of the at least one flag.

11. (Original) The apparatus of Claim 10 wherein the at least one instruction is at least one of: an integer instruction, an arithmetic instruction and a logical instruction.

12. (Original) The apparatus of Claim 11 wherein the at least one instruction includes at least one input operand.

13. (Original) The apparatus of Claim 10 including a flag register operatively coupled to the controller, wherein the controller updates a flag in the flag register if the flag modification enable bit is set to allow modification of the flag in the flag register.

14. (Cancelled) The apparatus of Claim 10 including a variable length instruction emulator that uses fixed length native instructions as the at least one instruction, to emulate variable length instructions.

15. (Currently amended) The apparatus of Claim 10 wherein the at least one native instruction is a fixed length instruction.

16. (Cancelled) The apparatus of Claim 14 wherein the variable length instruction emulator emulates non-native instructions using native instructions containing the flag modification bit.

17. (Previously presented) The apparatus of Claim 10 wherein the non-native instructions are variable length X86 instructions.

18. (Original) The apparatus of Claim 17 including:

an instruction converter, operatively coupled to receive variable length X86 instructions, that converts the received variable length X86 instructions to a plurality of native instructions wherein the plurality of native instructions include the at least one flag modification enable bit set to allow changing of non-native instruction flags in response to execution of the plurality of native instructions; and

a non-native instruction emulator operatively responsive to an unconvertible instruction command generated in response to detecting that an X86 instruction is not convertible by the converter, wherein the non-native instruction emulator emulates unconverted variable length X86 instructions using a plurality of native instructions wherein the native instructions include the at least one flag modification enable bit set to prevent changing of non-native instruction flags in response to execution of the plurality of native instructions for the unconverted variable length instruction.

19. (Currently amended) A method for processing program instructions comprising the steps of:

receiving at least two non-native instructions containing data representing operational code and at least one non-native instruction flag;

emulating the at least two non-native instructions using at least one native instruction including a flag modification enable bit;

determining whether the at least one flag modification enable bit allows updating of at least one flag register associated with the at least one non-native instruction flag in response to emulating the at least two non-native instructions; and

updating the at least one flag register in response to determining whether the at least one flag modification enable bit allows updating of the at least one flag.

20. (Currently amended) The method of claim 19 including preventing updating of the at least one flag register in response to determining whether the at least one flag modification enable bit allows updating of the at least one flag.